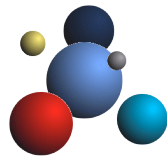


12 de Julio de 2002

Sistemas Computacionales en Bioinformática

Ignacio Martín Llorente
www.dacya.ucm.es/nacho

UCM



Grupo de Arquitectura de Sistemas Distribuidos y Seguridad
Departamento de Arquitectura de Computadores y Automática
Universidad Complutense de Madrid



Laboratorio de Computación Avanzada y Simulación
Centro de Astrobiología CSIC/INTA
Asociado al NASA Astrobiology Institute

¿Cuáles son las necesidades de computación en Bioinformática?



¿Qué alternativas de supercomputación existen?

¿Qué modelos de programación podemos utilizar?

¿Cuál es el futuro de la computación de altas prestaciones?

Proporcionar una visión global y práctica del estado actual y de las últimas tendencias en computación de altas prestaciones y su aplicación en Bioinformática

IMPORTANCIA DE LOS SISTEMAS DE ALTAS PRESTACIONES

- ✓ Simulación numérica sobre plataformas de altas prestaciones
- ✓ ¿Es realmente necesaria la computación paralela?
- ✓ Obstáculos en la paralelización de un código
- ✓ Propiedades numéricas y arquitectónicas de los métodos numéricos
- ✓ Ejemplos de necesidad de la computación de altas prestaciones en bioinformática

PRESENTE DE LA SUPERCOMPUTACIÓN

- ✓ Tipos de arquitecturas paralelas
- ✓ Paradigmas de programación
- ✓ Modelos de programación
- ✓ Experiencia personal

FUTURO DE LA SUPERCOMPUTACIÓN

- ✓ Problemas actuales
- ✓ Grid Computing
- ✓ Ejemplos

¿Cuál es la utilidad de la
computación de altas
prestaciones?



- Utilidad de la Simulación Numérica

Evita tener que construir prototipos

- Disminuye costes (*aeronáutica*)
- Aumenta la productividad al disminuir el tiempo de desarrollo (*aeronáutica*)
- Investigación más segura (*centrales nucleares*)

Permite realizar simulaciones que de otro modo no se podrían realizar

- Sistemas muy grandes (*Astrofísica*)
- Sistemas muy pequeños (*Dinámica molecular*)

Permite estudiar la validez de un modelo matemático

caro: aerodinámica
grande: cosmología
pequeño: diseño de fármacos
complejo: dinámica de partículas
peligroso: central nuclear
lento: aerodinámica

- Necesidad de la Potencia de Cálculo
-

Estudio de fenómenos a escala microscópica

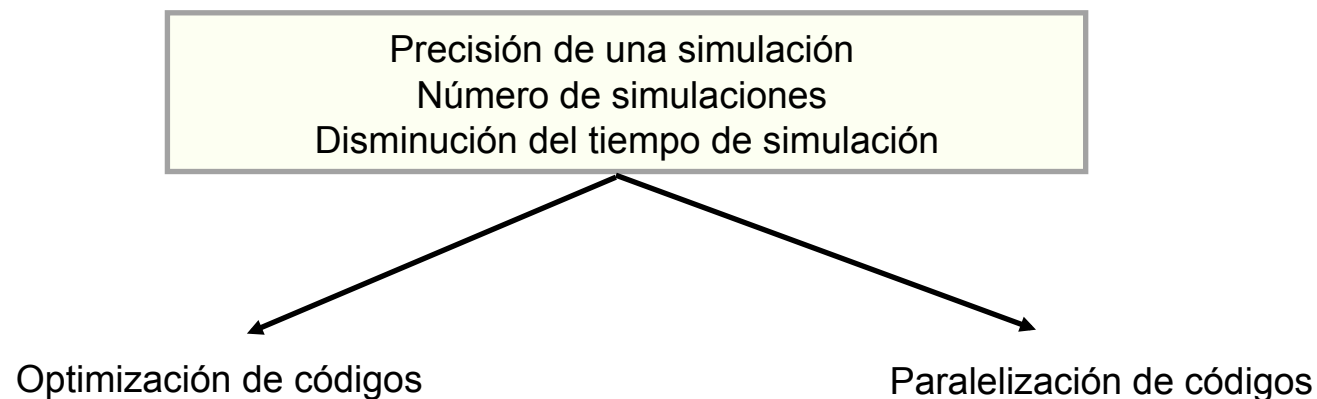
(dinámica de partículas)

- Resolución limitada por la potencia de cálculo del computador
- Cuantos más grados de libertad (puntos), mejor reflejo de la realidad

Estudio de fenómenos a escala macroscópica

(sistemas descritos por ecuaciones diferenciales fundamentales)

- Precisión limitada por la potencia de cálculo del computador
- Cuantos más puntos, más se acerca la solución discreta a la continua



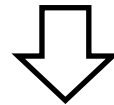
- Posibilidades de la Potencia de Cálculo

- Resolución de un problema en menor **tiempo de ejecución**, usando mas procesadores (**crítico en Tiempo**)
- Resolución de un problema con mayor **precisión**, usando mas memoria (**crítico en Precisión**)
- Resolución de un problema más real, usando modelos matemáticos más **complejos** (**crítico en Complejidad**)
- Resolver el mismo problema múltiples veces de forma paramétrica (**crítico en Productividad**)

Ley de Moore

- La velocidad de los microprocesadores se multiplica por cuatro cada 3 años

¿Para qué programar en paralelo?, podemos esperar a que mi estación sea lo suficiente potente



- Los microprocesadores aplican las técnicas de los computadores paralelos
- Cada vez es más difícil acercarse al pico (*jerarquía de memoria*)

Velocidad de la luz

- Computador con 1Tflop y 1 TB (1 THz)
- La distancia r para recoger el dato de memoria $<c/10^{12} = 0.3$ mm

Tamaño del computador 0.3x0.3



1 palabra de memoria ocupa 3 amstrons x 3 amstrons = tamaño de 1 átomo

Cierto

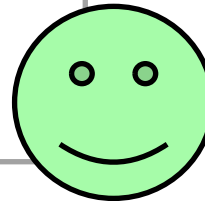
- ✓ El programador tiene que cambiar de forma de pensar
- ✓ Los algoritmos que solemos usar son secuenciales

Falso

- ✓ Es muy costoso programar los computadores paralelos existentes
- ✓ No existen herramientas de depuración y de optimización de código
- ✓ Los códigos paralelos no son portables

Propiedades Numéricas

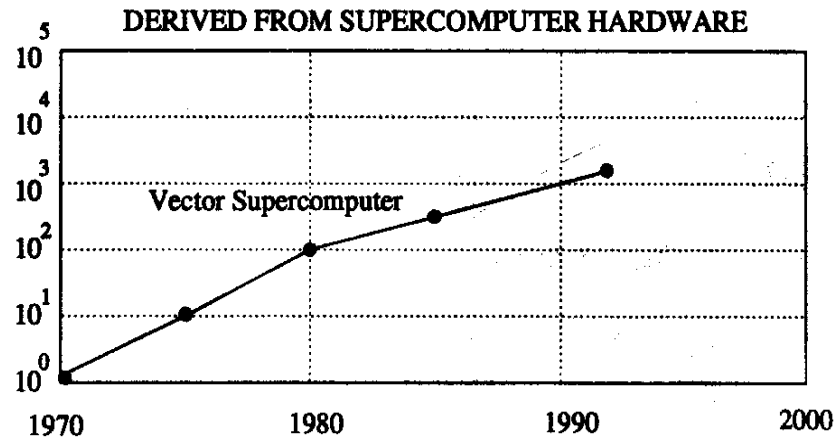
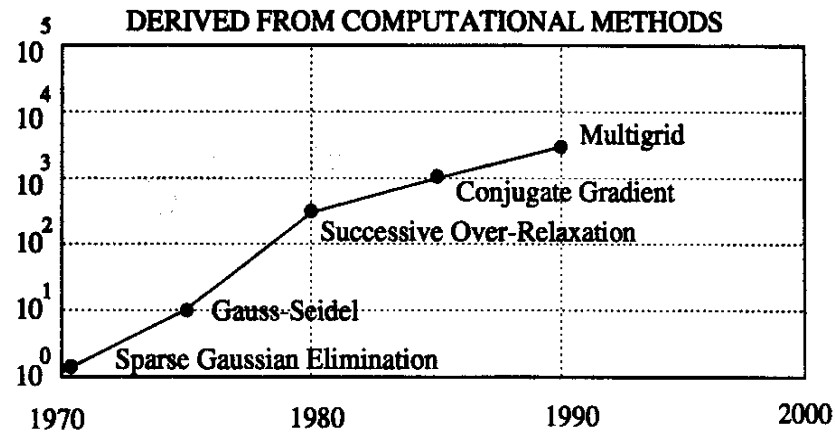
- ✓ Complejidad numérica del algoritmo



Otras Propiedades del Código
• Inversión (fácil adaptación, portabilidad, ...)

Propiedades Arquitectónicas

- ✓ Consumo de memoria
- ✓ Localidad de los datos => Explotación de la jerarquía de memoria
- ✓ Grado de paralelismo => Explotación de múltiples procesadores



Grand Challenge: High Performance Computing and Communications (NSF)

“I would rather have today’s algorithms on yesterday’s computers than viceversa”

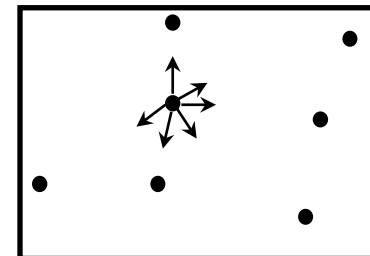
Toint

• Importancia de las Propiedades Numéricas

Problema:

- Conjunto de N moléculas sometidas a la ley de Newton
- El potencial es la suma de potenciales de corto y largo alcance

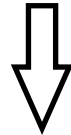
P	$O(N^2)$	$O(N \log N)$
	MOLMEC 7.000	MEGADYN 550.000
1	8152	
2	4481	6305
3	3956	
4	2427	3295
6	1769	
8		1849



- Ambos tienen speedups lineales
- La simulación de 550.000 partículas consumiría más de 18.000 procesadores con MOLMEC

- Métodos Multiescala

“Un punto en un dominio físico requiere progresivamente menos información y de modo menos frecuente de otras partes del dominio que están progresivamente más alejadas”

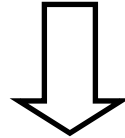


Métodos Multimalla
Métodos Multinivel
Métodos Adaptativos
.
.
.

Multigrid Net
Instituto Weizmann
ICASE (NASA Langley)
GMD

www.mgnet.org
www.weizmann.ac.il
www.icasel.edu
www.gmd.de

Resolución de problemas independientes



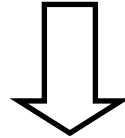
Alta Productividad

Ejemplo: Comparación de secuencias (BLAST)

- Proporcionando una secuencia determinada busca en una base de datos de secuencias aquellas que alcancen cierto umbral según una matriz de comparación
- Información proporcionada:
 - ✓ Ayuda a identificar proteínas relacionadas evolutivamente
 - ✓ Teniendo proteínas caracterizadas experimentalmente permite realizar predicciones sobre la función de otras proteínas
- Coste de operaciones suponiendo:
 - ✓ 3.000 genes por genoma bacteriano
 - ✓ 60 genomas
 - ✓ 180.000 secuencias

www.ncbi.nlm.nih.gov/BLAST/

Resolución de un problema en menos tiempo



Alto Rendimiento

Ejemplo: Ejecución paralela de las comparaciones de secuencias (BLAST multithreading)

Ejemplo: Generación de alineamientos múltiples (CLUSTALW)

- Después de realizar una comparación de secuencias se definen clusters de secuencias que son parecidas dentro de un cierto umbral
- Estas herramientas generan alineamientos de todas las secuencias por medio de matrices de comparación y un esquema de penalización específico

`www.ebi.ac.uk/clustalw/`

Ejemplo: Generación de perfil del grupo de secuencias (HMMER)

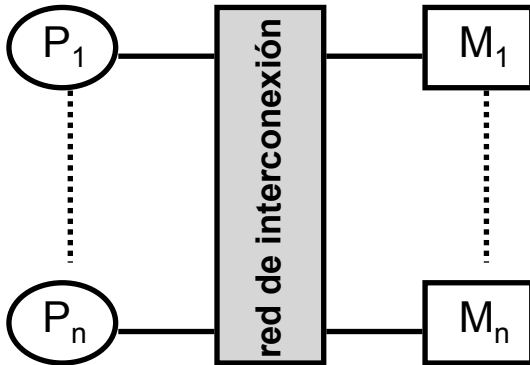
- Partiendo de un alineamiento múltiple genera un perfil del grupo de secuencias por medio del modelo de las cadenas de Markov para posteriormente buscar en la base de datos secuencias que se adaptan al perfil

`hmmmer.wustl.edu/`



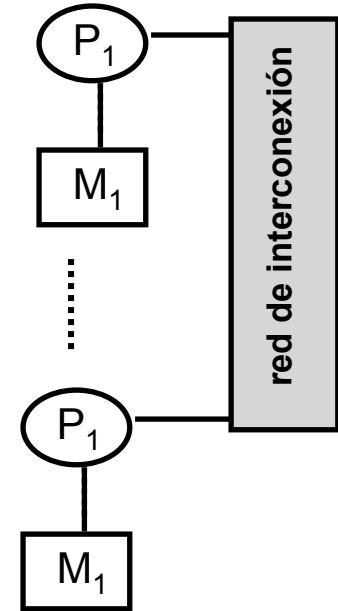
¿Qué arquitecturas existen actualmente y cómo se programan?

Memoria Compartida

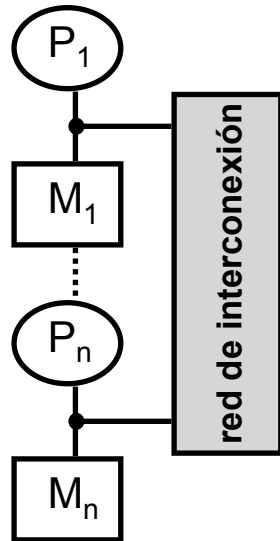


Programación vs. escalabilidad

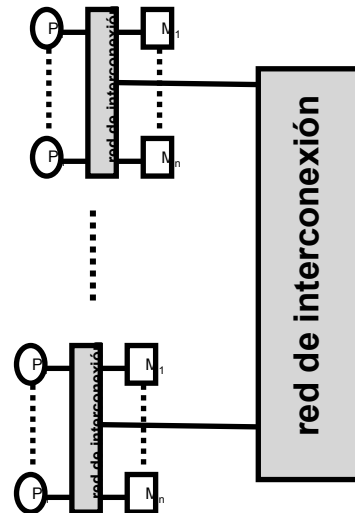
Memoria Distribuida



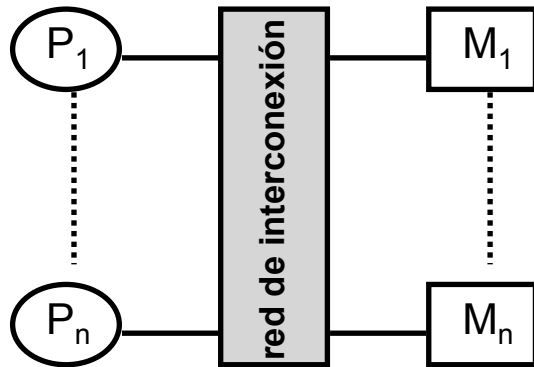
UMA
(Uniform Memory Access)
SMP



NUMA
(Non Uniform Memory Access)



Cluster de UMAs



UMA
 (Uniform
 Memory
 Access)
SMP

SUN E10.000 Starfire (256 cpus; 137 Gflops)

- 64 UltraSparcII 400 MHz
- punto-a-punto 1,6 GB/seg, crossbar (12,8 GB/seg)
- 51,2 Gflops y 64 Gbytes



HP 9.000 SuperDome (128 cpus; 196 Gflops)

- 64 PA Risc 8600 552 MHz
- punto-a-punto 4 GB/seg, crossbar (64 GB/seg)
- 141 Gflops y 128 Gbytes



COMPAQ AlphaServer GS320 (8 cpus; 11,8 Gflops)

- 32 Alpha 21264 731 MHz
- punto-a-punto 6,4 GB/seg, crossbar (51,2 GB/seg)
- 46,8 Gflops y 256 Gbytes



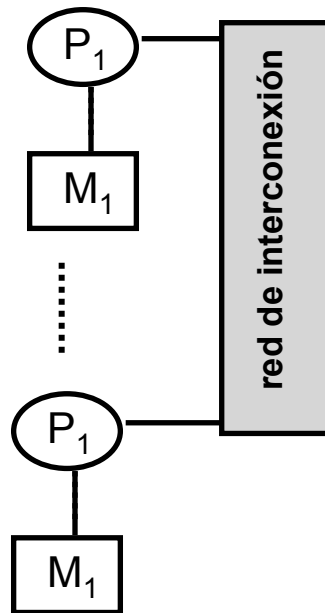
Fujitsu/Siemens PrimePower 2000 (128 cpus; 102 Gflops)

- 128 UltraSparcII 563 MHz
- punto-a-punto 1,6 GB/seg, crossbar (12,8 GB/seg)
- 144 Gflops y 4 Tbytes



www.serverworldmagazine.com
www.top500.org/ORSC/2001/

- Memoria Distribuida

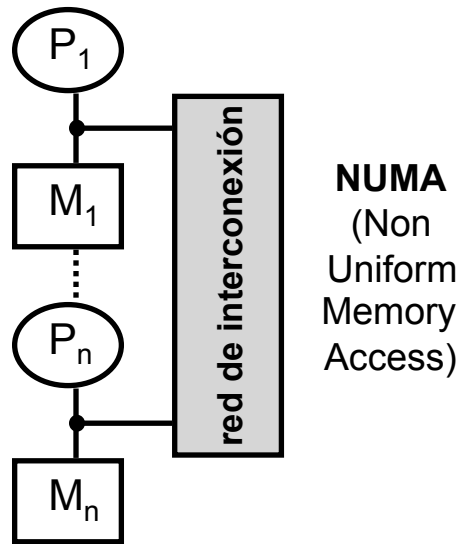


Cray T3E 1350 (1488 cpus 600MHz; 1.127 Tflops)

- 2176 Alpha 21164 675 MHz
- punto-a-punto 325 MB/seg, toro
- 2,9 Tflops y 1 Tbyte



www.serverworldmagazine.com
www.top500.org/ORSC/2001/



NUMA
(Non
Uniform
Memory
Access)

SGI Origin 3800 (ASIC Blue 6144 cpus; 1,6 Tflops)

- 512 MIPS R14.000 500MHz
- punto-a-punto 1,6 GB/seg, hipercubo
- 512 Gflops y 1 Tbyte



SUN Fire E15.000

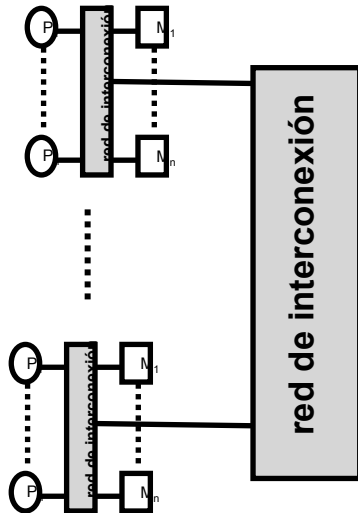
- 72 (106) UltraSparcIII 900 MHz
- punto-a-punto 3,2 GB/seg, 3xcrossbar (172,8 GB/seg)
- 190 Gflops y 576 Gbytes



Programación vs. escalabilidad

www.serverworldmagazine.com
www.top500.org/ORSC/2001/

Memoria virtualmente compartida y físicamente distribuida



IBM RS/6800 SP (ASCI White 8.000 cpus; 7,2 Tflops)

- 2048 (128x16) Power3 375 MHz
- SMP punto-a-punto 500 MB/seg, crossbar (222 GB/seg)
- MD [punto-a-punto 500 MB/seg, omega]
- 3.07 Tflops y 4 Tbytes



COMPAQ AlphaServer SC (3.024 cpus 1 GHz; 4,0 Tflops)

- 512 (32x16) Alpha 21264 833 MHz
- SMP[punto-a-punto 6,4 GB/seg, crossbar (51,2 GB/seg)]
- MD[punto-a-punto 210MB/seg, fat tree]
- 853 Gflops y 2 Tbytes



www.serverworldmagazine.com
www.top500.org/ORSC/2001/

Linpack performance.netlib.org/performance/html/linpack.data.col0.html

Computer (Full Precision)	Number of Processors	R_{max} Gflop/s	N_{max} order	$N_{1/2}$ order	R_{peak} Gflop/s
ASCI White-Pacific, IBM SP Power 3(375 MHz)	8000	7226	518096	179000	12000
Compaq AlphaServer SC ES45/1GHz	3024	4059	525000	105000	6048
Compaq AlphaServer SC ES45/1GHz	3000	3643	525000	140000	6000
IBM SP Power3 208 nodes 375 MHz	3328	3052.	371712		4992
IBM SP Power3 158 nodes 375 MHz	2528	2526.	371712	102400	3792
ASCI Red Intel Pentium II Xeon core 333MHz	9632	2379.6	362880	75400	3207
ASCI Blue-Pacific SST, IBM SP 604E(332 MHz)	5808	2144.	431344	432344	3868
ASCI Red Intel Pentium II Xeon core 333MHz	9472	2121.3	251904	66000	3154

Top 500 www.top500.org

List	Rank	Manufacturer	Computer	R_{max} (GFlops)	Installation Site	Country	Year	Processors	R_{peak} (GFlops)	N_{max}	N_{half}
11/2001	1	IBM	ASCI White, SP Power3 375 MHz	7226.00	Lawrence Livermore National Laboratory	USA	2000	8192	12288	518096	179000
11/2001	2	Compaq	AlphaServer SC ES45/1 GHz	4059.00	Pittsburgh Supercomputing Center	USA	2001	3024	6048	525000	105000
11/2001	3	IBM	SP Power3 375 MHz 16 way	3052.00	NERSC/LBNL	USA	2001	3328	4992	371712	102400
11/2001	4	Intel	ASCI Red	2379.00	Sandia National Labs	USA	1999	9632	3207	362880	75400
11/2001	5	IBM	ASCI Blue-Pacific SST, IBM SP 604e	2144.00	Lawrence Livermore National Laboratory	USA	1999	5808	3868	431344	
11/2001	6	Compaq	AlphaServer SC ES45/1 GHz	2096.00	Los Alamos National Laboratory	USA	2001	1536	3072	390000	71000
11/2001	7	Hitachi	SR8000/MPP	1709.10	University of Tokyo	Japan	2001	1152	2074	141000	16000
11/2001	8	SGI	ASCI Blue Mountain	1608.00	Los Alamos National Laboratory	USA	1998	6144	3072	374400	138000

- **Sistemas de Bajo Coste (*Intranet Computing*)**

⇒ Utilización de una biblioteca estándar de paso de mensajes (PVM, MPI, ...) para programar una red de estaciones como si se tratara de un computador paralelo de memoria compartida

Ventajas:

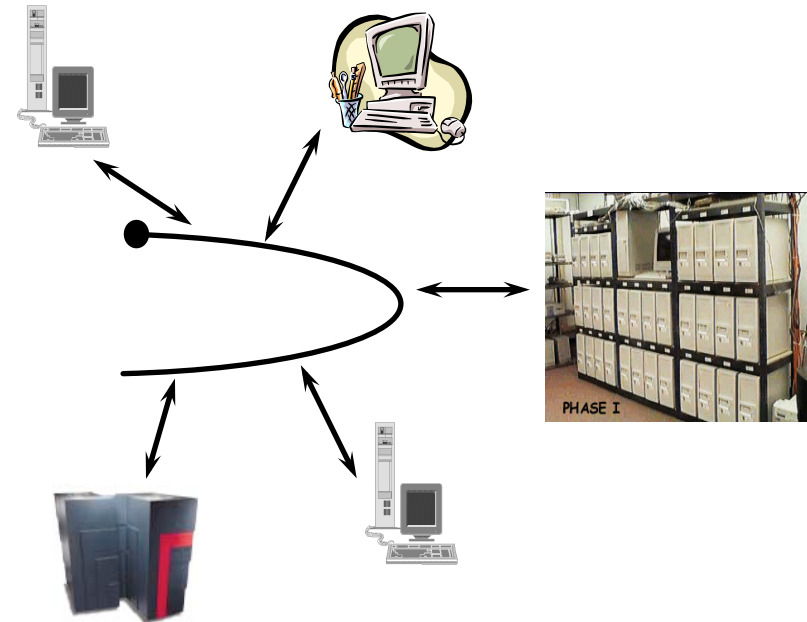
- Gratis

Inconvenientes:

- Coste de las comunicaciones muy alto
 - Bus lento
 - Acceso secuencial al bus
 - Bus compartido con otras aplicaciones
 - Sobrecarga TCP/IP
- La carga de las estaciones varía dinámicamente
- Las estaciones de trabajo tienen velocidades diferentes

Conclusión:

- Es una buena solución solo para aplicaciones con grano muy grueso



- Sistemas de Bajo Coste (*Intranet Computing*)
-

Software disponible

Sun Grid Engine de Sun Microsystems

www.sun.com/gridware

LSF de Platform Computing

www.platform.com

Condor de la Universidad de Wisconsin

www.cs.wisc.edu/condor

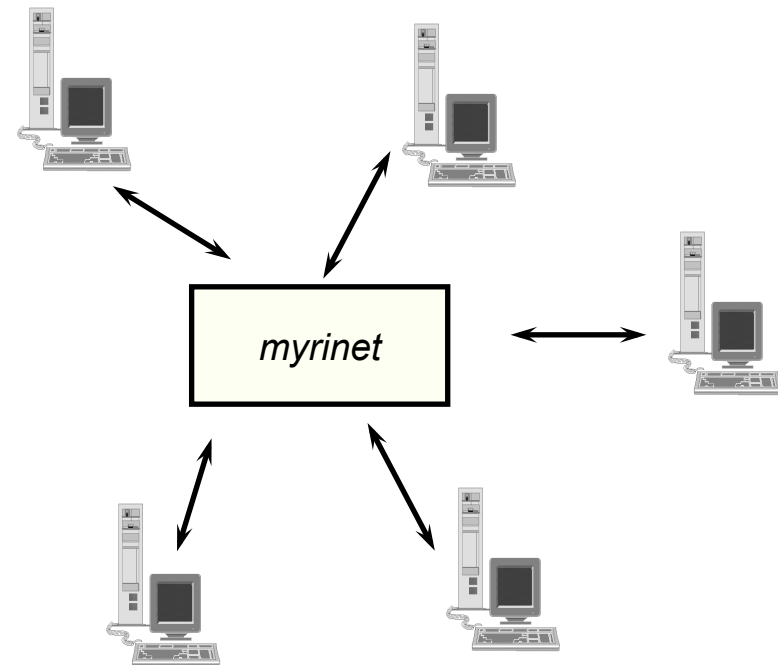
PBS de Veridian Systems

www.openpbs.org

- **Sistemas de Bajo Coste (*Cluster Computing*)**

⇒ Un sistema Beowulf es un cluster de estaciones o computadores personales homogéneo dedicado a computación paralela

⇒ Suele estar dotada de una red avanzada basada en *router* con *Fast Ethernet* o *Myrinet*



- **Sistemas de Bajo Coste (*Cluster Computing*)**

Front-end: Alpha DS20 a 500 Mhz

Nodos de trabajo: 30 nodos Alpha DS10 a 466 Mhz

Memoria RAM: 8 Gb

Disco: 300 Gb.

Rendimiento pico: 30 Gflops

Red

- **Servicios:** Fast-Ethernet (100Mbits/seg)
- **Comunicación:** ServerNet II (1Gbit/seg)

Sistema Operativo: Alpha 7.0 de Suse

Software de gestión de carga:PBS

Librerías de paralelización: MPI

Gestión de usuarios: NFS y NIS

Babioca



http://dalbe.inta.es/~LCASAT/trab/o_babioca.htm

- Sistemas de Bajo Coste (*Cluster Computing*)

¿Cómo implementar una máquina paralela de bajo coste?

HispaCluster

www.hispacluster.org

IEEE TFCC (reports)

www.ieeetfcc.org

Ejemplos

Beowulf Project en CESDIS

www.beowulf.org

Avalon

cnls.lanl.gov/Internal/Computing/Avalon/

Sandia Labs Computational Plant

www.cs.sandia.gov

Coral en ICASE

www.icasel.edu

Babieca en el CAB

dalbe.inta.es/~LCASAT/trab/o_babieca.htm

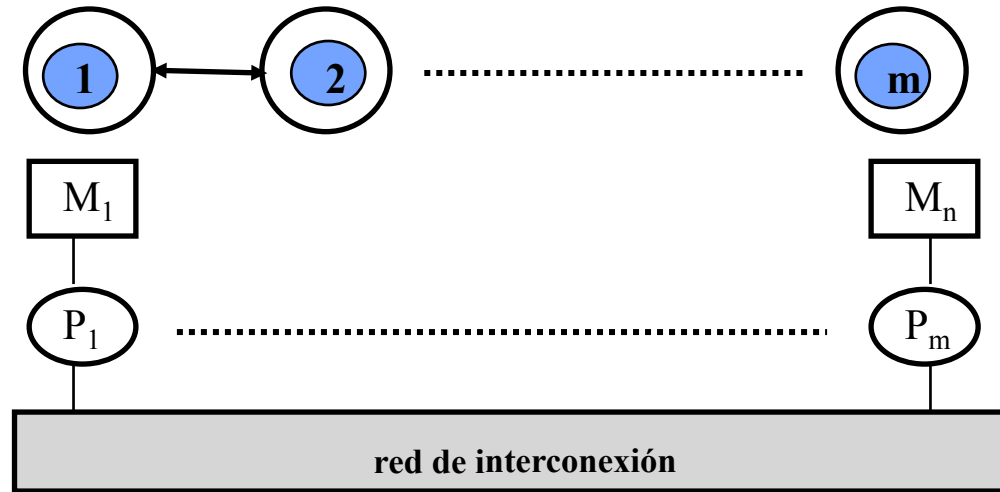
Software

- Herramientas de distribución de carga, NFS, NIS...
- Interesante el software Mosix de la Universidad de Israel (www.mosix.cs.huji.ac.il)

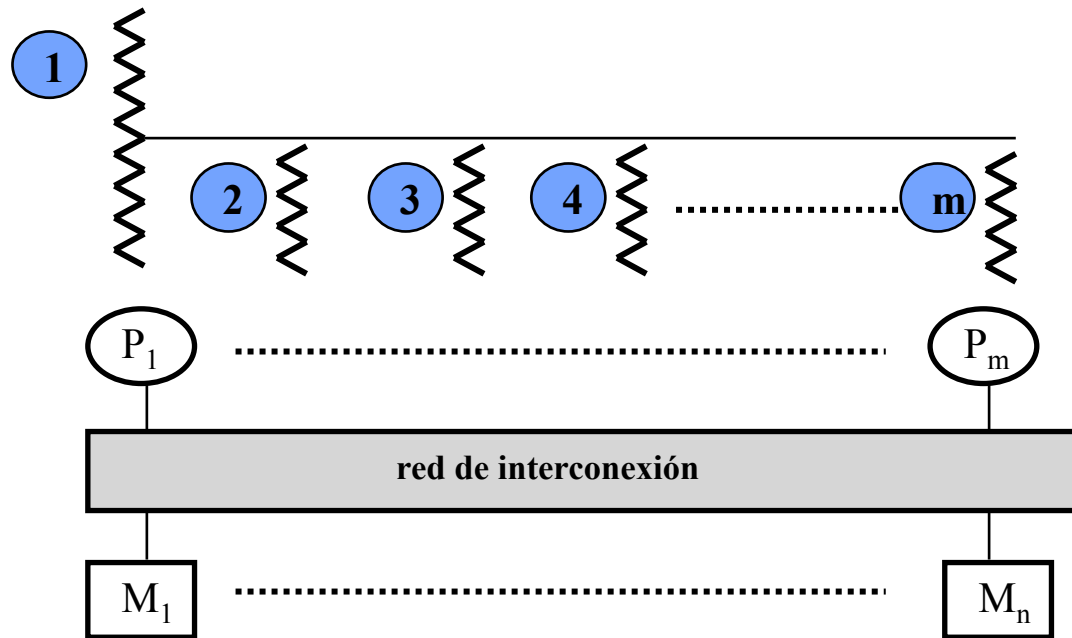
“Computer science is no more about computers than astronomy is about telescopes”

W. Dijkstra

Memoria Distribuida



Memoria Compartida

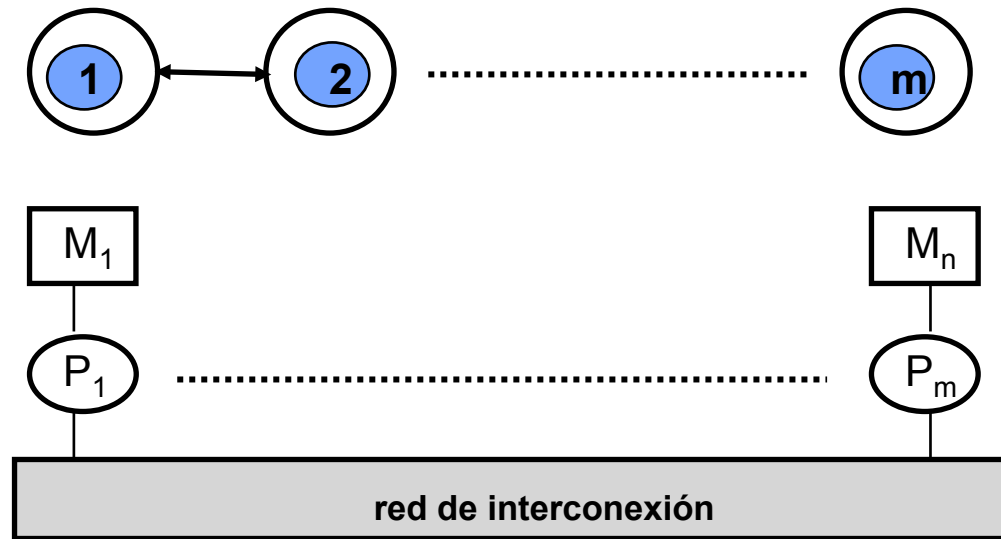


Programación por medio de mensajes

- Comunicación por paso de mensajes
- Primitivas de gestión de comunicaciones
 - Envío
 - Recepción
 - Sincronización

MPI
PVM
Librerías nativas

Nivel superior:
(directivas de distribución de datos)
HPF



MPI www.mcs.anl.gov/mpi/
PVM www.epm.ornl.gov/pvm/pvm_home.html
HPF www.crpc.rice.edu/HPFF

- **MPI (Message Passing Interface)**

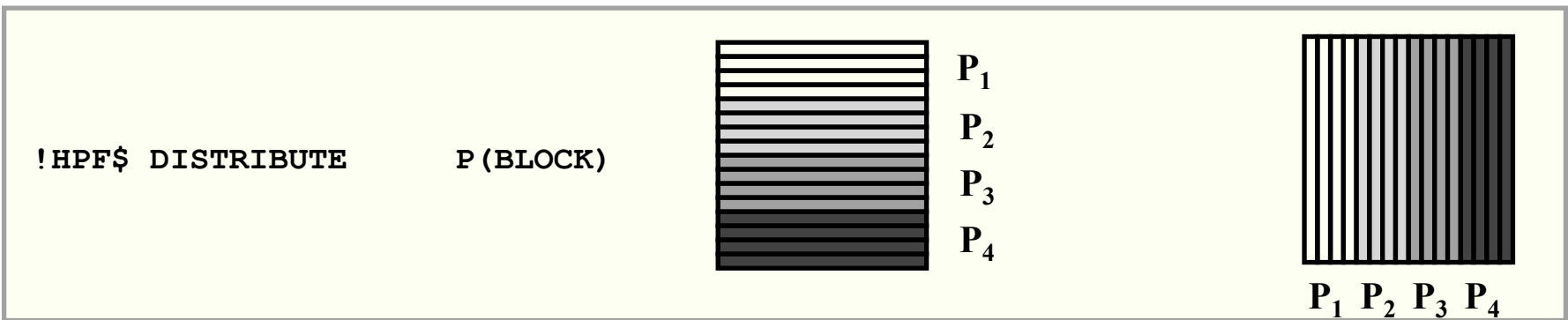
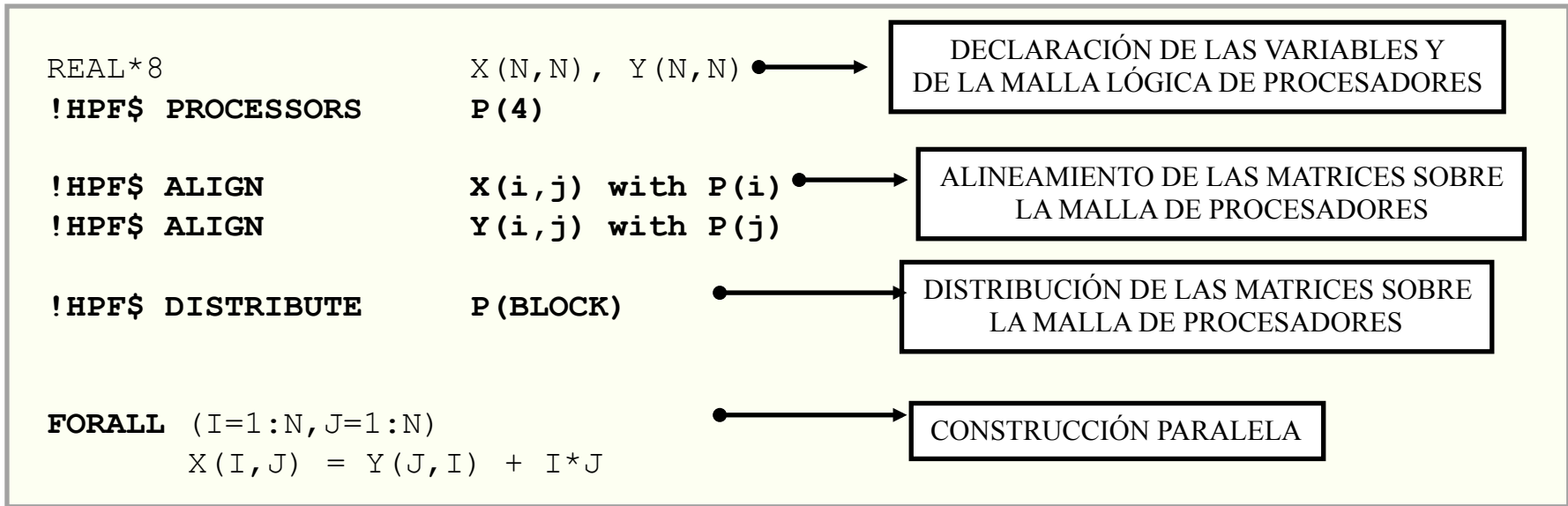
Estructura de un Programa (Solo 6 rutinas)

```
#include <stdio.h>
#include <mpi.h>

main(int argc, char **argv)
{
    int mi_rango, numero_procesos, etiqueta=50, destino=0, origen;
    char mensaje[100];
    MPI_Status estado;
    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &mi_rango);
    MPI_Comm_size(MPI_COMM_WORLD, &numero_procesos);

    if (mi_rango !=0) {
        sprintf(mensaje,"¡Saludos desde el proceso %d!", mi_rango);
        MPI_Send(mensaje, strlen(mensaje)+1, MPI_CHAR, destino, etiqueta, MPI_COMM_WORLD);
    } else {
        for (origen = 1; origen < numero_procesos; origen++) {
            MPI_Recv(mensaje, strlen(mensaje)+1, MPI_CHAR, origen, etiqueta, MPI_COMM_WORLD, &estado);
            printf("%s\n", mensaje);
        }
    }
    MPI_Finalize();
}
```

Estructura de un Programa (Descomposición de datos)



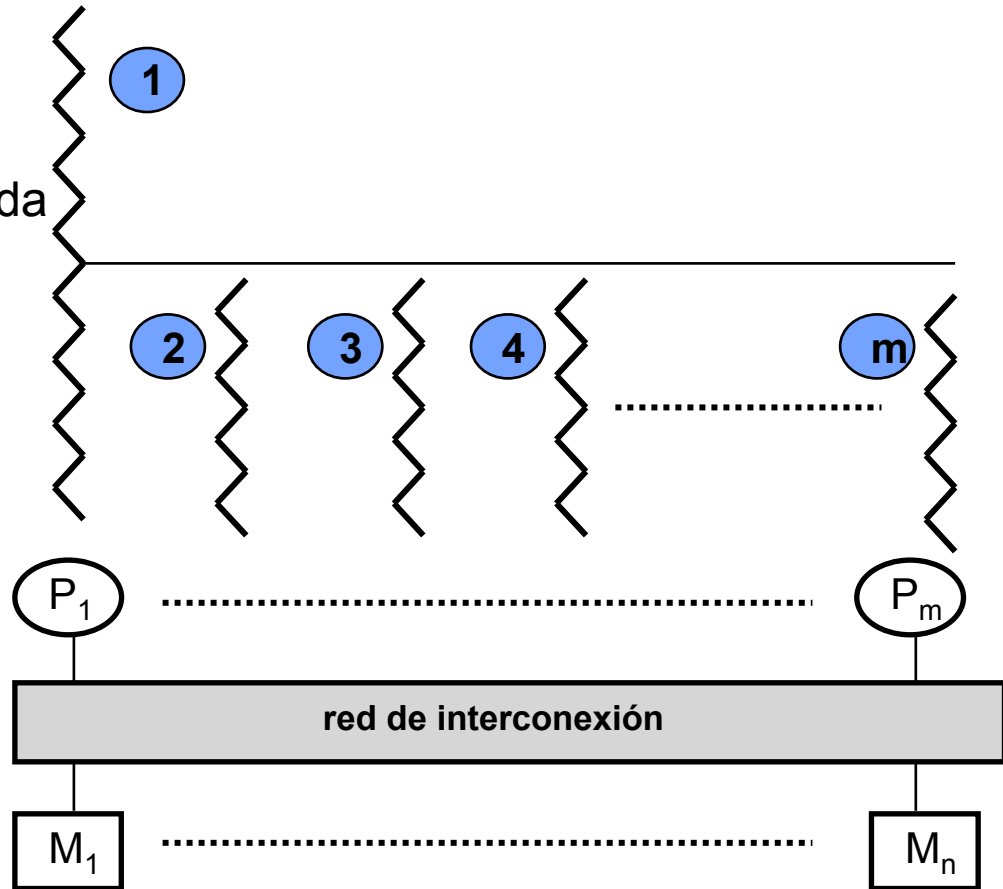
• Paradigma de Memoria Compartida

Programación por medio de *threads*

- Comunicación por memoria compartida
- Primitivas de gestión de threads
 - Creación
 - Espera
 - Protección de datos

Posix Threads
 Librerías nativas

Nivel superior:
 (directivas de distribución de datos)
 DOACROSS
 DOALL
 X3H5
OPENMP



Threads www.humanfactor.com/pthreads/pthread-tutorials.html
OpenMP www.openmp.org

• Posix Threads

```
void *print_message_function( void *ptr );
pthread_mutex_t mutex;
main()
{
    pthread_t thread1, thread2;
    pthread_attr_t pthread_attr_default;
    pthread_mutexattr_t pthread_mutexattr_default;
    struct timespec delay;
    char *message1 = "Hello";
    char *message2 = "World\n";

    delay.tv_sec = 10;
    delay.tv_nsec = 0;

    pthread_attr_init(&pthread_attr_default);
    pthread_mutexattr_init(&pthread_mutexattr_default);

    pthread_mutex_init(&mutex, &pthread_mutexattr_default);
    pthread_mutex_lock(&mutex);

    pthread_create( &thread1, &pthread_attr_default,
                   (void *) print_message_function, (void *) message1);
    pthread_mutex_lock(&mutex);
    pthread_create(&thread2, &pthread_attr_default,
                  (void *) print_message_function, (void *) message2);
    pthread_mutex_lock(&mutex);
    exit(0);
}
```

```
void *print_message_function( void *ptr )
{
    char *message;
    message = (char *) ptr;
    printf("%s ", message);
    pthread_mutex_unlock(&mutex);
    pthread_exit(0);
}
```

• Directivas



Problemas de la programación por medio de threads

- Análisis del código
- Recodificación
- Incluir primitivas *multithreading*

No cambiamos el lenguaje pero ...



Paralelización por medio de directivas

- Ejecución de un **bucle** sobre múltiples procesos

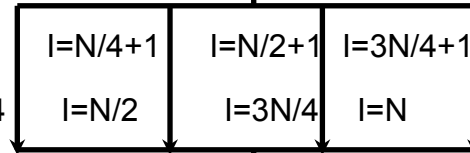
```

C$PAR DOALL
  DO I=1, N
    A(I) = B(I)
  END DO

```

Bucle Paralelo

I=1



I=N/4

Fin Bucle Paralelo

Sección Secuencial

Sección Secuencial

Nivel de abstracción superior:

- Menos flexible
- Más portable
- Menos eficiente
- Más rápido

• Comparación

Paso de Mensajes (MPI):

- ☹ Difícil de programar
- ☹ No soporta paralelización incremental de un código (partiendo del puro secuencial)
- ☺ Portabilidad
- ☺ Rendimiento

HPF:

- ☹ No es tan utilizado como inicialmente se estimó
- ☹ Tiene bastantes limitaciones (paralelismo en control)
- ☹ Los compiladores son todavía deficientes
- ☺ Sencillez

Posix Threads:

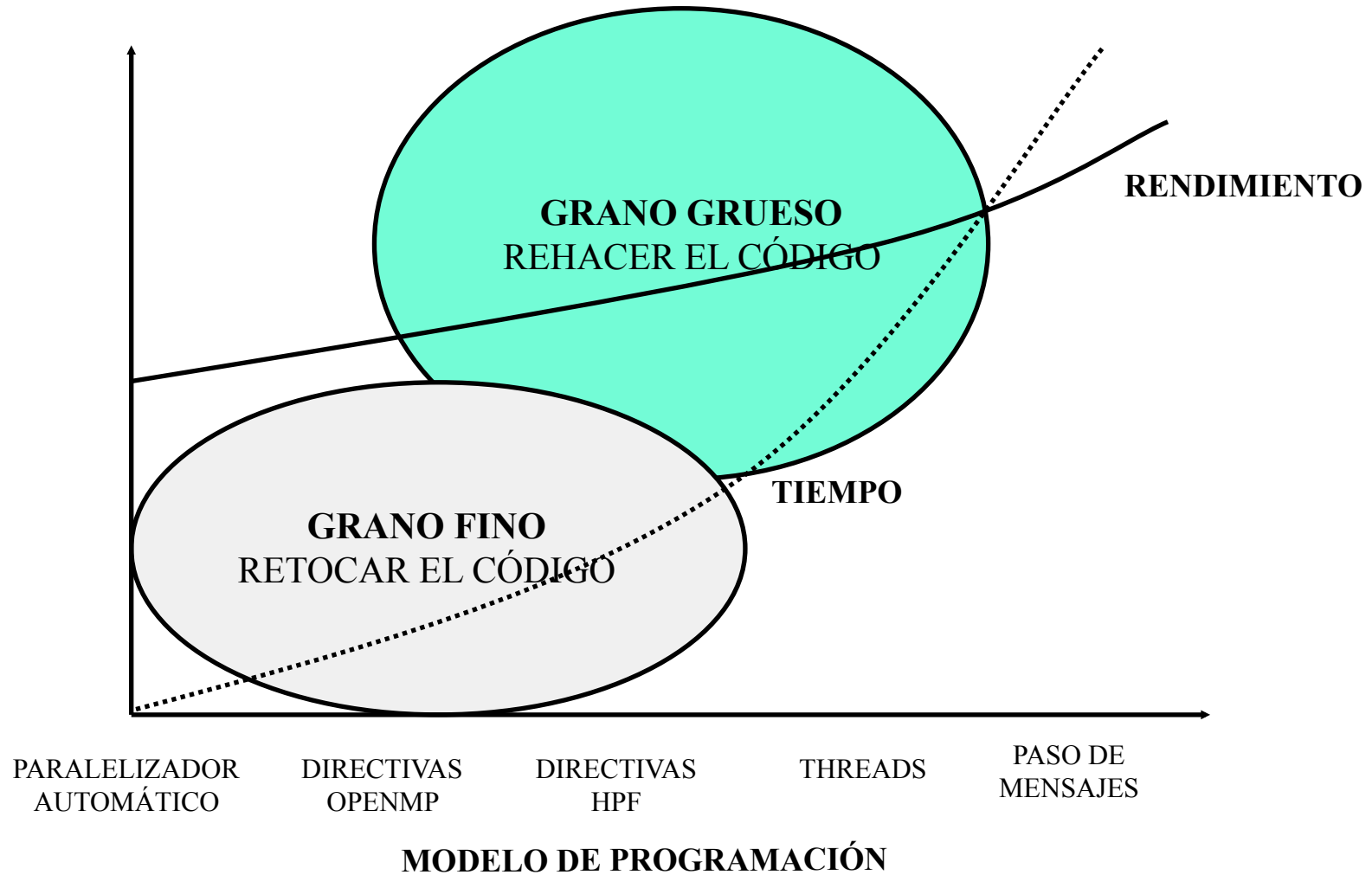
- ☹ Complejos
- ☹ No son utilizados por la computación numérica
- ☹ Prácticamente sin soporte para Fortran 77
- ☹ Incluso en lenguaje C exige programar a demasiado bajo nivel
- ☺ Modelo usado para herramientas de sistema

Directivas:

- ☹ Únicamente válido para memoria compartida
- ☹ Rendimiento poco escalable
- ☺ Sencillo
- ☺ Soporta paralelismo incremental

LA SOLUCIÓN DEPENDE DE
(TIEMPO, RENDIMIENTO)

PARALELIZAR
VS.
DESARROLLAR UN CÓDIGO PARALELO



La mayoría de las veces es suficiente con usar directivas

- La mayoría de problemas se pueden resolver con pocos procesadores en un computador de memoria compartida
- La mayoría de las máquinas que existen son de memoria compartida (véase el Top 500)
- La mayoría de las veces se quiere mejorar un código secuencial que ya existe de forma incremental y en poco tiempo a pesar que el rendimiento obtenido no sea el mejor

Justificación del uso de paso de mensajes

- Portabilidad
- En sistemas Beowulf
- En clusters de estaciones de trabajo
- En computación masivamente paralela
- Siempre que el problema requiera realmente potencia de calculo



¿Cuál es el futuro de la
supercomputación ?

*“It’s hard to make predictions,
especially about the future”*

Yogi Berra

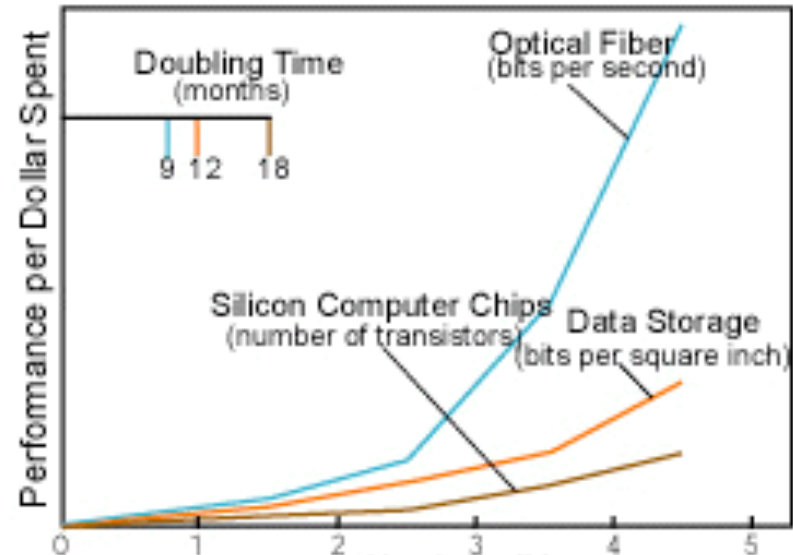
Necesito más potencia de cálculo

- La potencia de los supercomputadores solo crece linealmente (dos órdenes de magnitud cada 10 años) (www.top500.org)

Muchos centros sufrirán este problema en pocos años

*La capacidad de almacenamiento se dobla cada 12 meses
 El ancho de banda de red se dobla cada 9 meses
 El rendimiento de un procesador se dobla cada 18 meses*

- 1986 to 2000
 - Computers: x 500
 - Networks: x 340,000
- 2001 to 2010
 - Computers: x 60
 - Networks: x 4000



Moore's Law vs. storage improvements vs. optical improvements. Graph from *Scientific American* (Jan-2001) by Cleo Vilett, source Vined Khoslan, Kleiner, Caufield and Perkins.

Conclusiones:

*Un único sistema no será capaz de analizar los datos que almacenen sus discos
 Un único centro no podrá analizar el volumen de información generado
 La red permitirá de forma eficiente usar recursos distribuidos
 (1 orden de magnitud de diferencia entre procesamiento y red)*

Los equipos son caros y difíciles de mantener

- ¿Cómo puedo amortizar la inversión realizada?
- ¿Qué hago si tengo un pico de demanda de cálculo y no tengo presupuesto para adquirir un supercomputador?

Desequilibrio de carga en el tiempo

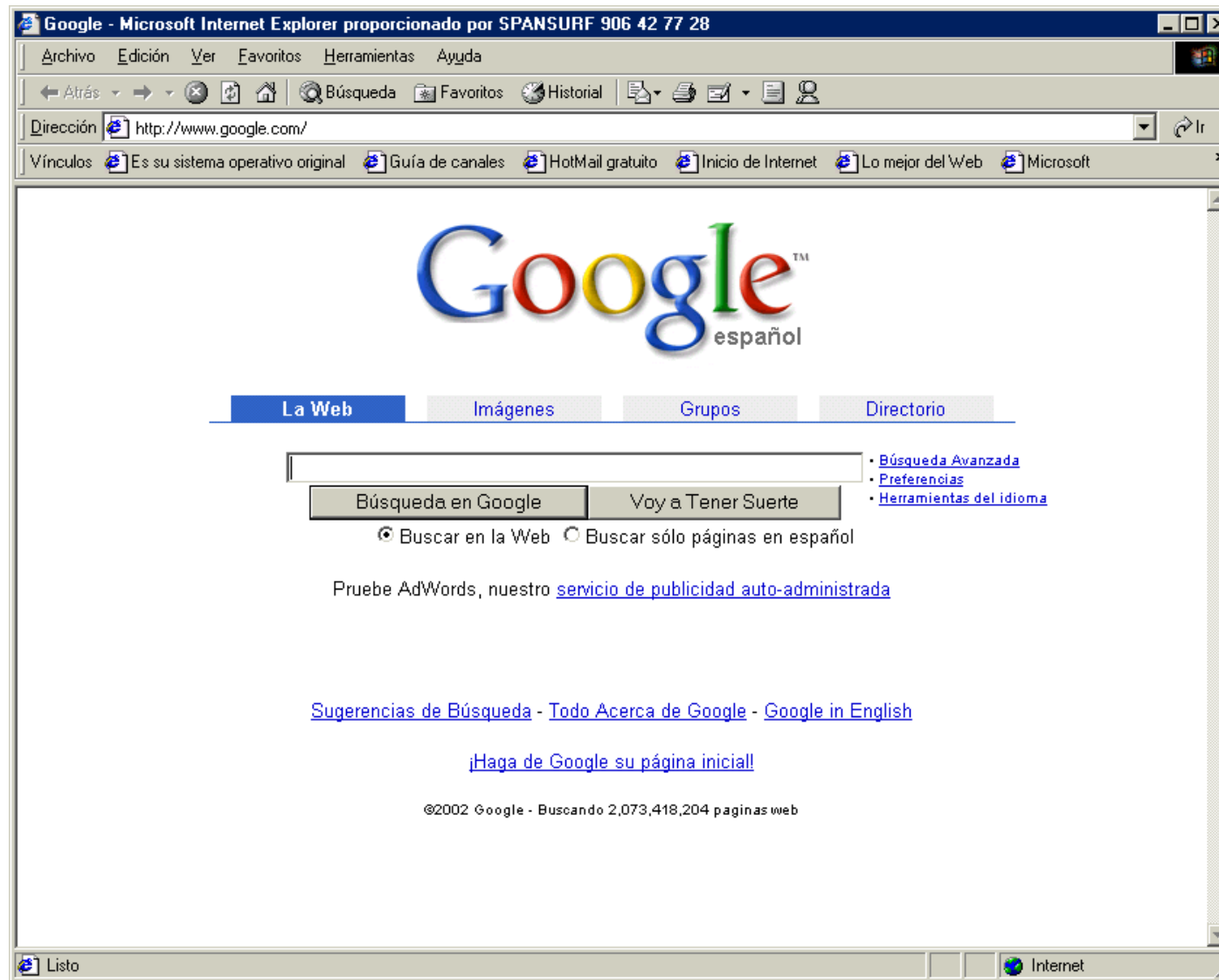
- ¿Cómo puedo conseguir una carga más homogénea?

Los sistemas PC y estaciones de la red están ociosos

- ¿Cómo puedo utilizar esta potencia desaprovechada?

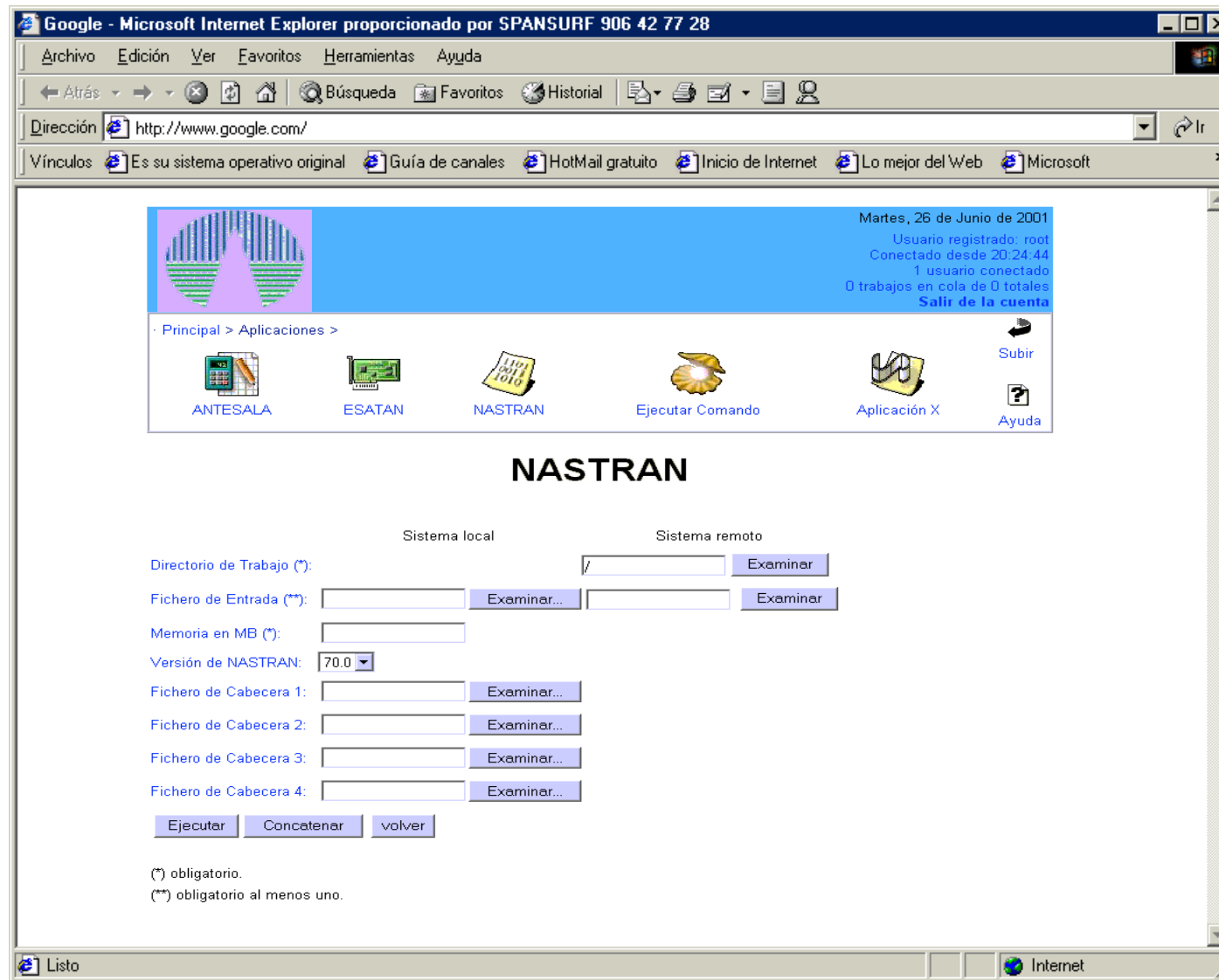


¿Qué hacemos cuando queremos
buscar información ?

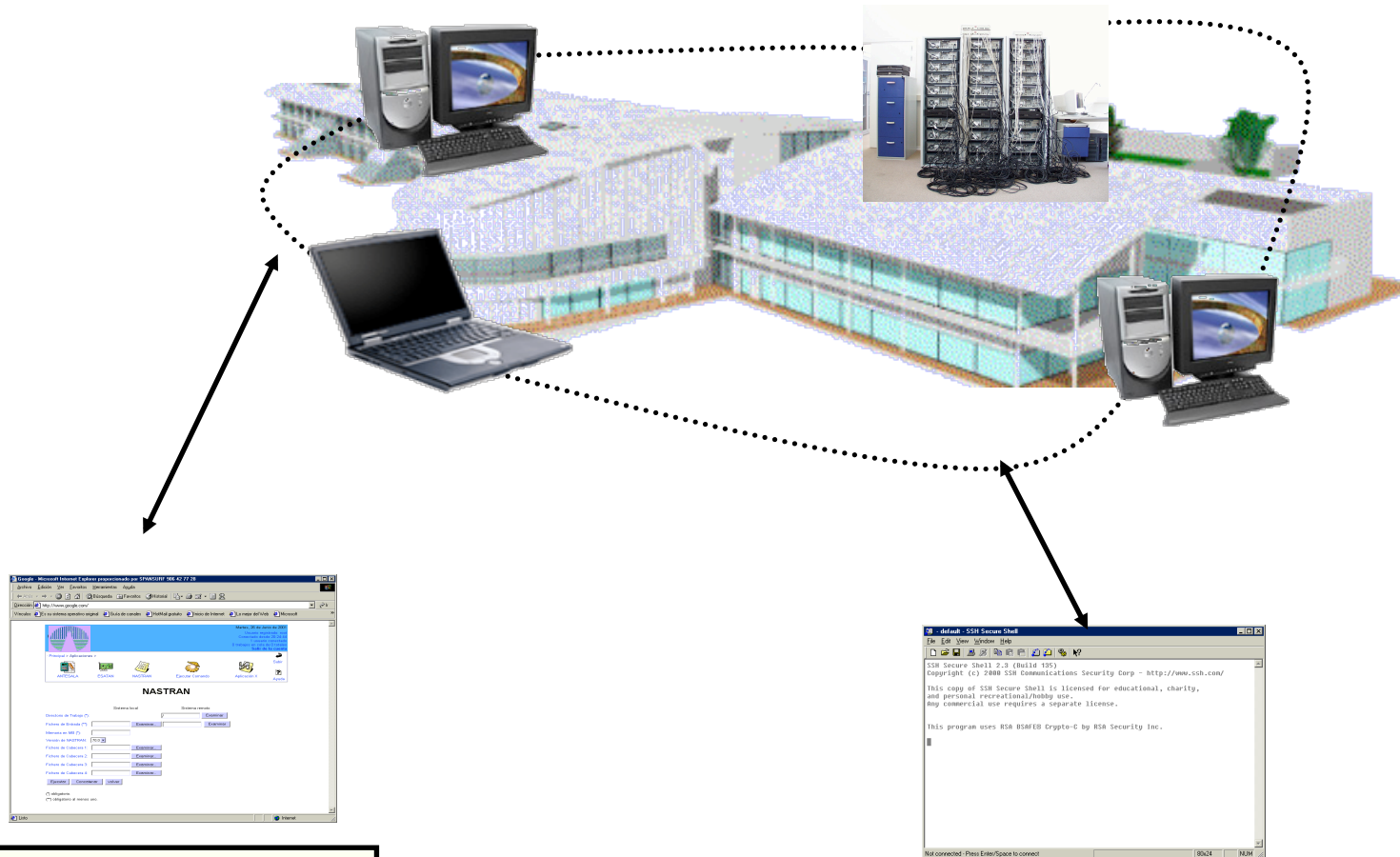




¿Qué haremos en el futuro cuando necesitamos potencia de cálculo ?



Grid interno \approx *Intranet Computing*

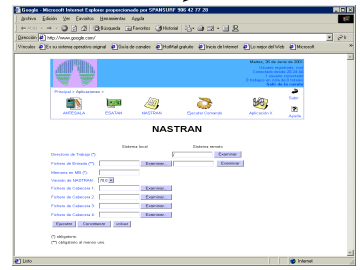


Computing Portal

Grid global



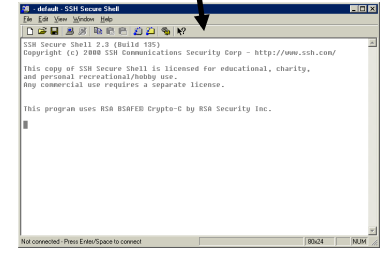
¿cuántos equipos se conectan al día a internet?



Computing Portal

Scale 1:134,000,000
Robinson Projection
standard parallels 38°N and 38°S

January 2002
Boundary representation is
not necessarily authoritative.
802804AI (R00352) 12-01





“Cuando Internet sea tan rápido como los buses internos de un computador, éste se desintegrará en la red en un conjunto de recursos de propósito específico”

Gilder Technology Report, Junio 2000



Imaginemos...

Ejecutar nuestra aplicación sobre todos los
supercomputadores del mundo

Interconectar los sensores de datos con
recursos de procesamiento y posterior
visualización en tiempo real

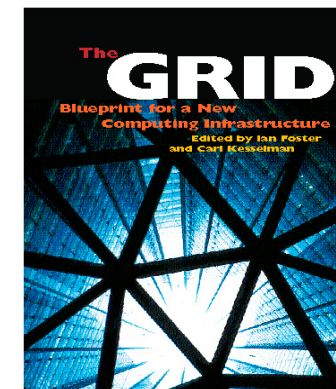
....y todo esto desde casa ...

- ⇒ Nueva tecnología cuyo objetivo es la compartición de recursos en Internet de forma uniforme, transparente, segura, eficiente y fiable

- ⇒ Análoga a las redes de suministro eléctrico:
 - Ofrecen un único punto de acceso a un conjunto de recursos distribuidos geográficamente (supercomputadores, clusters, almacenamiento, fuentes de información, instrumentos, personal, bases de datos...)

- ⇒ Los sistemas distribuidos se pueden emplear como un único sistema virtual (metacomputador u organización virtual) en aplicaciones intensivas en datos o con gran demanda computacional.

The Grid : Blueprint for a New Computing Infrastructure
Ian Foster y Karl Kesselman
Morgan Kaufmann Publishers, 1998



Objetivos

- ⇒ Conseguir un aprovechamiento óptimo de los recursos distribuidos
- ⇒ Convertir Internet, total o parcialmente, en un "único" sistema

Implicaciones

- ⇒ Mayor grado de colaboración entre sistemas interconectados por medio de la compartición transparente de todo tipo de recursos
- ⇒ Nuevas líneas de investigación en seguridad, sistemas de información, planificación de recursos, sistemas de ficheros distribuidos, portales de acceso...

Desafíos técnicos

- ⇒ Recursos heterogéneos
- ⇒ Diferentes modalidades del mismo recurso (arquitecturas, sistemas operativos...)
- ⇒ Descubrimiento, selección, reserva, asignación, gestión y monitorización de recursos
- ⇒ Desarrollo de aplicaciones
- ⇒ Desarrollo de modelos eficientes de uso
- ⇒ Comunicación lenta y no uniforme

Desafíos socioeconómicos

- ⇒ Organizativos: Dominios de administración, modelo de explotación y costes, política de seguridad...
- ⇒ Económicos: Precio de los recursos, oferta/demanda...

Ciencia e Ingeniería

- ⇒ Alquiler de recursos
- ⇒ Amortización de recursos
- ⇒ Gran potencia de cálculo a precio bajo sin necesidad de adquirir equipamiento
- ⇒ Mayor colaboración y compartición de recursos entre varios centros

Empresas

- ⇒ Alquiler de recursos
- ⇒ Organizaciones virtuales
- ⇒ Negocios basados en proveer recursos

Motivación económica:
Acceso a cualquier recurso sin adquirirlo

• Investigación

Infraestructura Grid: Servicios y protocolos básicos para interconectar recursos

- ⇒ Globus: www.globus.org
- ⇒ Legion: www.cs.virginia.edu/~legion/
- ⇒ Polder: www.science.uva.nl/research/scs/PSCS4.html
- ⇒ MOL: www.uni-paderborn.de/pc2/projects/mol/

Toolkits de aplicación: Módulos para construir aplicaciones Grid específicas

- ⇒ Nimrod/G: www.csse.monash.edu.au/~rajkumar/ecogrid/
- ⇒ Condor: www.cs.wisc.edu/condor/condorg
- ⇒ Data Grid: www.eu-datagrid.org
- ⇒ Portal: dast.nlanr.net/Projects/GridPortal
- ⇒ MPI/G: www.globus.org/mpi

Aplicaciones Grid

- ⇒ CACTUS (www.cactuscode.org)
- ⇒ Virtual Laboratory (www.csse.monash.edu.au/~rajkumar/vlab/)

Bancos de prueba Grid: Sistemas Grid para prototipos y producción

- ⇒ *NASA's Information Power Grid:* www.nas.nasa.gov/About/IPG/ipg.html
- ⇒ *European Data Grid:* www.eu-datagrid.org
- ⇒ *Grid Physics Network:* www.griphyn.org
- ⇒ *Teragrid:* www.teragrid.org
- ⇒ *Tidewater Research Grid:* www.tidewaterrgp.org

Enlaces sobre el Grid

Recopilación de enlaces

www.gridcomputing.com

Global Grid Forum

www.gridforum.org

Últimas noticias

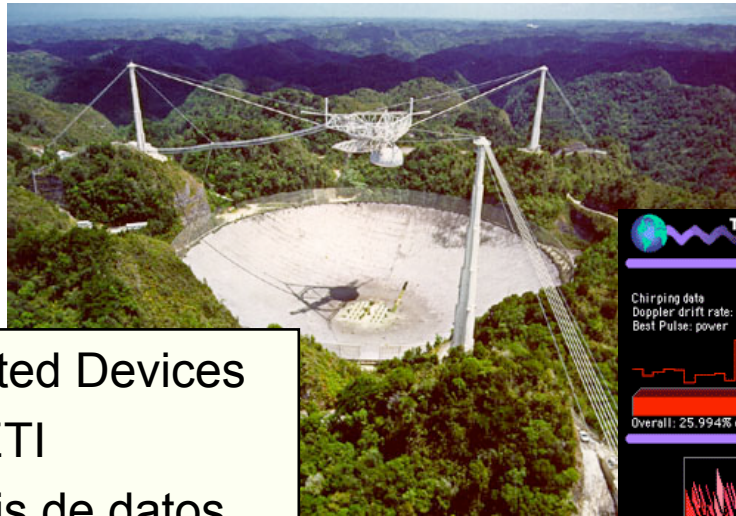
www.gridcomputingplanet.com

PCs domésticos para avanzar en la lucha contra el SIDA

Comunidad: 1000 PCs
Tecnología: Entropia
Laboratorio: Scripps
Objetivo: Avanzar en la investigación contra el SIDA (*docking*: predecir a nivel molecular como un fármaco aplicado a una proteína puede contrarrestar el efecto de un virus)

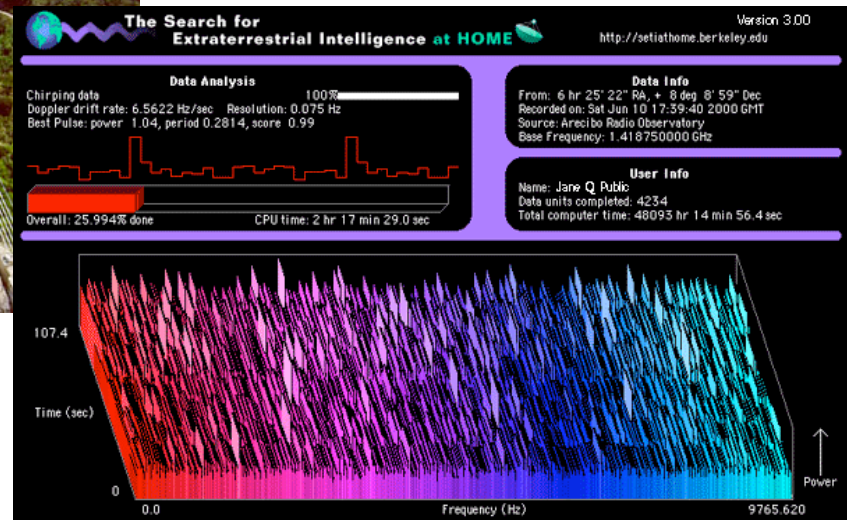
www.fightaidsathome.org

PCs domésticos para buscar vida extraterrestre



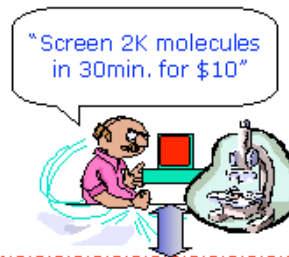
setiathome.ssl.berkeley.edu

Tecnología: United Devices
Laboratorio: SETI
Objetivo: Análisis de datos de telescopio (Arecibo, Puerto Rico) en búsqueda de señales



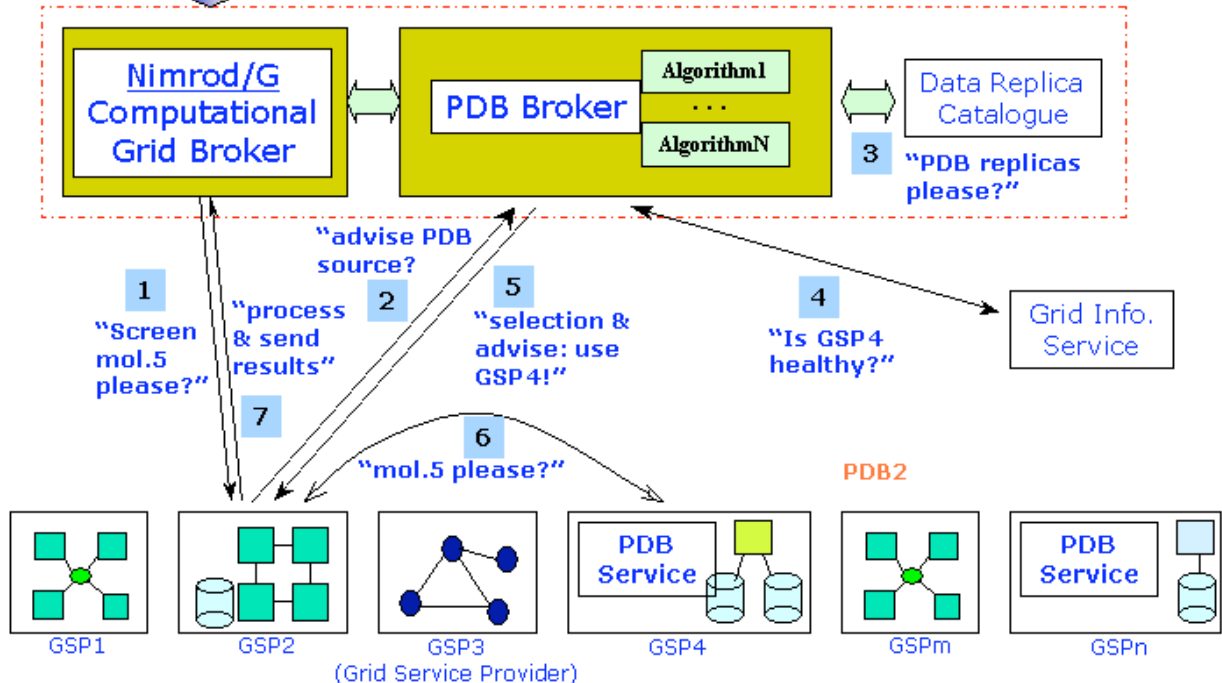
	Total	Last 24 Hours
Users	3555143	1659
Results received	451999430	465542
Total CPU time	890267.654 years	837.087 years
Floating Point Operations	1.346983e+21	1.815614e+18 (21.01 TeraFLOPs/sec)
Average CPU time per work unit	17 hr 15 min 13.9 sec	15 hr 45 min 04.6 sec

Virtual Laboratory



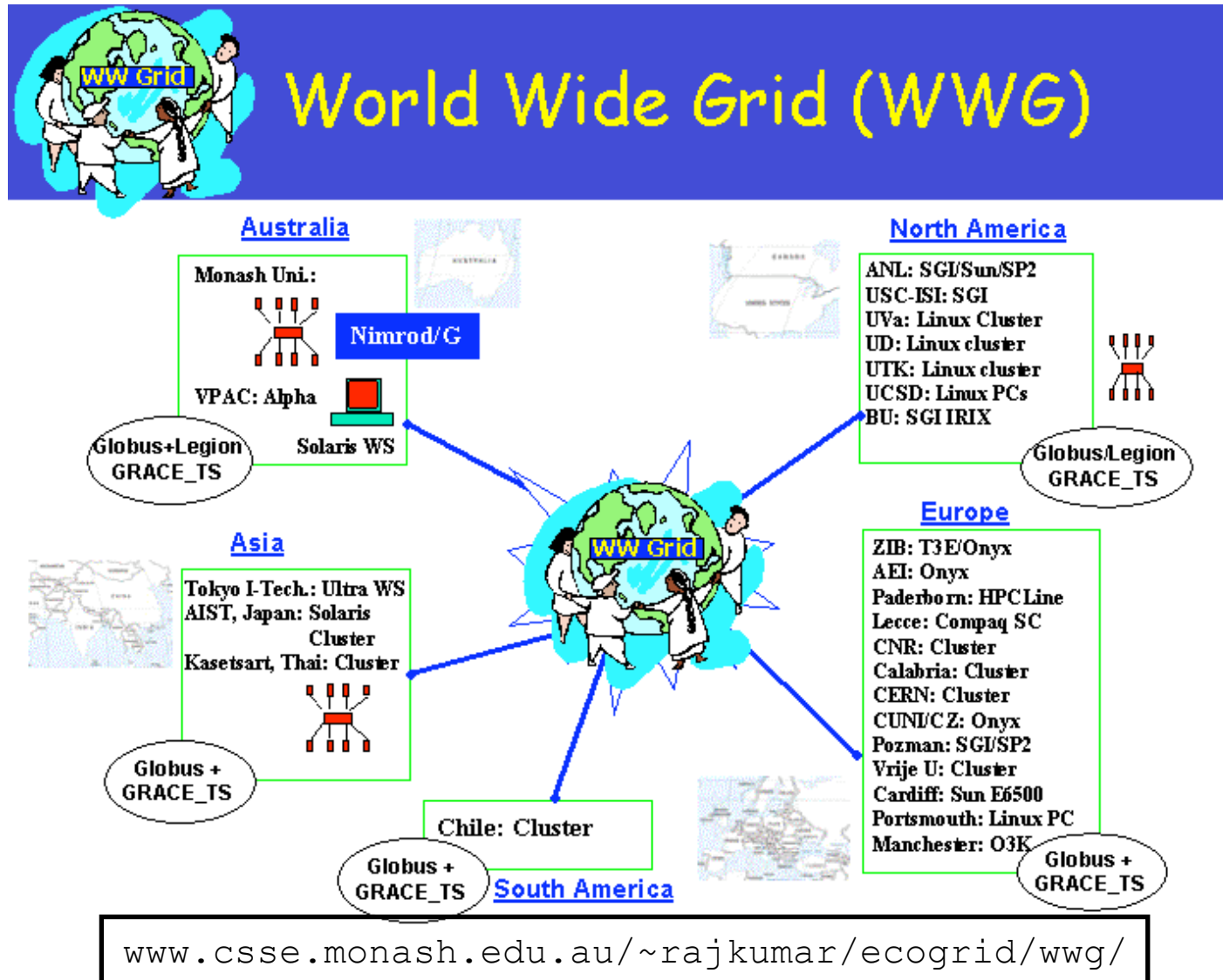
Resource Brokering Architecture for Molecular Screening on World Wide Grid

Comunidad: World Wide Grid
Tecnología: Nimrod/G + Globus
Laboratorio: WEHI
Objetivo: Resolución de problemas intensivos en datos y computación de biología molecular (*docking*)



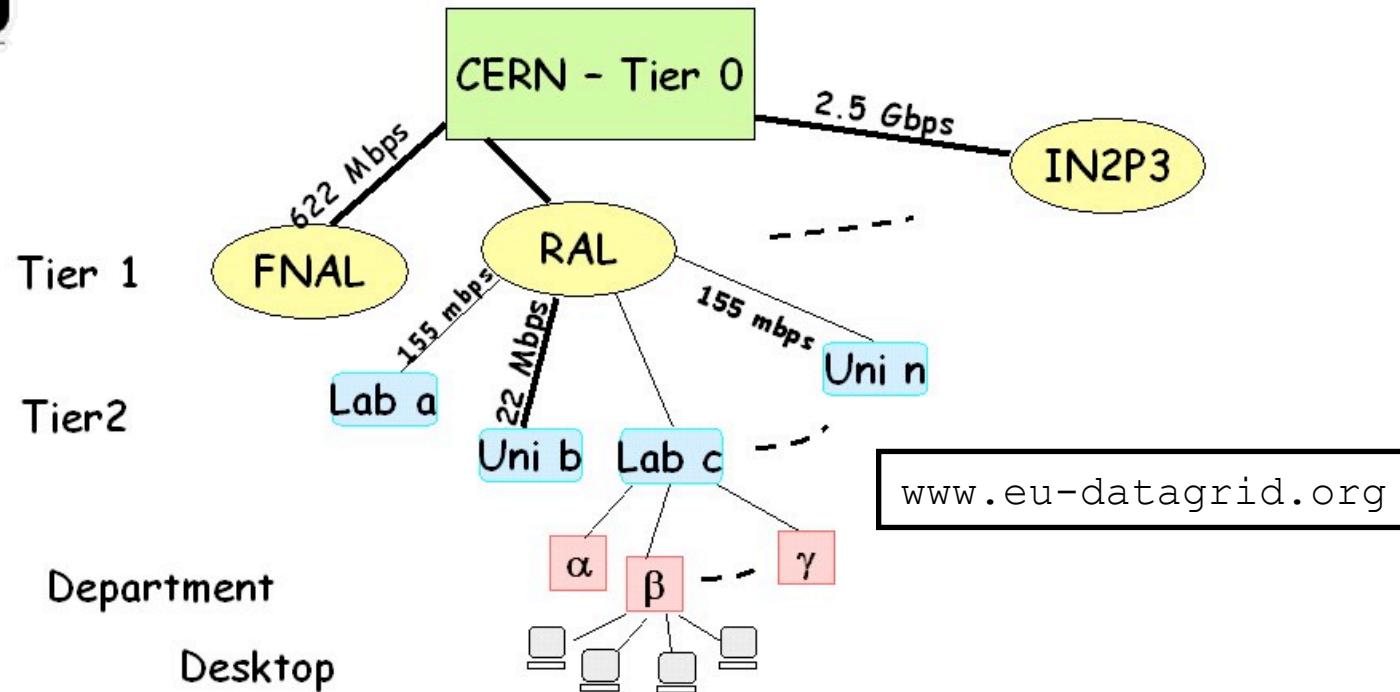
www.csse.monash.edu.au/~rajkumar/vlab/

Banco de pruebas





European Data Grid



Tecnología: European Data Grid + Globus

Laboratorio: CERN (ESA,CNRS,INFN,NIKHEF y PPARC)+asociados

Objetivo: Desarrollar la tecnología necesaria para permitir los requerimientos de computación y análisis de bases de datos compartidas, desde cientos de TeraBytes a PetaBytes, en la investigación del futuro

“Technology does not drive change at all. Technology merely enables change. It’s our collective cultural response to the options and opportunities presented by technology that drives change”

Paul Saffo